Remote Workstation Card Agent for Linux Administrators' Guide

23.06

Table of Contents

PCoIP Remote Workstation Card Agent for Linux

This guide is intended for administrators who are deploying the Remote Workstation Card Agent for Linux as part of

The PCoIP Remote Workstation Card Agent for Linux introduces Teradici brokering to a Teradici Remote Workstation Card deployment, allowing the desktop to be managed by Teradici Anyware Manager or by third-party brokers that support the PCoIP Broker protocol.

A complete PCoIP Remote Workstation Card deployment includes these components:

- **A physical host machine**, which provides the desktop to remote clients. See System Requirements for more information.
- **A PCoIP Remote Workstation Card** installed on the host machine.
- **The PCoIP Remote Workstation Card software for Linux** installed on the host machine.
- **The Remote Workstation Card Agent for Linux** installed on the host machine.

About PCoIP Licensing

When the Remote Workstation Card Agent for Linux is installed, the Remote Workstation Card can be licensed using a

What's New in This Release

Release 23.06 of the Remote Workstation Card Agent for Linux includes:

- This release maintains version parity with other products.

System Requirements

The Remote Workstation Card Agent for Linux depends on the following system capacities and capabilities:

Host Instance Requirements

Global instance requirements	
Operating Systems	RHEL/CentOS 7; RHEL or Rocky Linux 8
Remote Workstation Card Firmware	5.1.0+
Remote Workstation Card Software for Linux installed version	4.8.0+
Remote Host Memory	At least *2GB* of RAM is required on the host desktop. The agent should have at least *512MB* of available memory.
Remote Host CPUs	At least 2 CPUs are required on the host desktop. Processors must support Streaming SIMD Extensions (SSE) 4.2.
Network Ports	The following ports must be open on the host desktop: • TCP 443 • TCP 4172 • UDP 4172 • TCP 60443
Storage	At least 100MB for installation and 100MB for logging are recommended.
User	Cannot be root. You must create a user account for PCoIP connections.

Installation Guide

Installing the PCoIP Remote Workstation Card Agent for Linux on RHEL or CentOS

Before you proceed with installation, a few prerequisites must be met.

Prerequisites

These instructions assume you have already built the remote desktop machine, and that the machine meets the agent's requirements.

Before proceeding with Remote Workstation Card Agent for Linux installation, install a desktop environment. To install a desktop environment in RHEL or CentOS, use the following command:

```
sudo yum groupinstall 'Server with GUI'
```

A few other things to confirm before proceeding:

- SSH must be enabled.
- You must have a license registration code for the agent instance from Teradici (as part of a Teradici Cloud Access subscription).
- The desktop machine requires the following ports to be open: TCP 443, TCP 60443, TCP 4172, and UDP 4172.
- You must have super user (root) privileges and be able to issue `sudo` commands.
- If you are using a PCoIP Local License Server, you'll need to know it's URL and port numbers.

Important: Protect your license registration code

The license registration code you receive from Teradici is unique to your organization, and should be protected as you would any sensitive data.

Be careful that you do not inadvertently expose your registration code in forums or other public areas by pasting log messages without redacting sensitive information.

Installation Overview

Once your prerequisites are in place, you can proceed with installation. Here's a brief overview of the process:

1. Connect to the machine using SSH.
2. Install the PCoIP Agent.
3. If required, configure the agent software.
4. Disconnect the SSH session.
5. Connect to the desktop using a PCoIP client.

If you're ready to start, connect to your machine with an SSH client and proceed to install the Remote Workstation Card Agent for Linux.

Installing the Remote Workstation Card Agent for Linux on RHEL or CentOS

Important: Required ports will be automatically opened

The Remote Workstation Card Agent for Linux installer will add firewall exceptions for the following required PCoIP ports during installation: TCP 443, TCP 4172, UDP 4172, and TCP 60443.

To install the Remote Workstation Card Agent for Linux:

Before you begin, confirm that your Remote Workstation Card and Remote Workstation Card Software are properly installed.

1. Confirm that you can create a direct connection from a PCoIP Zero Client to the Remote Workstation Card machine. After verifying, disconnect the session.
2. Download and install the Teradici repository, via the shell script provided here.
3. Install the EPEL repository:

```
sudo yum install epel-release
```

4. Install the PCoIP Remote Workstation Card Agent for Linux:

```
sudo yum install pcoip-agent-standard
```

5. Note your machine's local IP address. Clients connecting directly to the host workstation will need this number to connect.
6. Enter the license registration code you received from us.

Note: These instructions are for Cloud Licensing

These instructions assume you are using Teradici Cloud Licensing to activate your PCoIP session licenses. If you are using the Teradici License Server instead, see Licensing the Remote Workstation Card Agent for Linux.

For unproxied internet connections, type:

```
pcoip-register-host --registration-code=<XXXXXX@YYY-YYYY-YYY>
```

For proxied internet conections, type:

```
pcoip-register-host --registration-code=<XXXXXX@YYY-YYYY-YYY> --proxy-
server=<serverURL> --proxy-port=<port>
```

7. Open `/etc/pcoip-agent/pcoip-agent.conf` with root privileges in a text editor.
8. Add the following line:

```
pcoip.server_type = "RWC"
```

9. Save the file and close the editor.
10. Reboot the desktop.

> **Note: About the package name**
>
> `pcoip-agent-standard` is the correct package for the Remote Workstation Card Agent.

2. License the Agent

The Remote Workstation Card Agent for Linux must be assigned a valid PCoIP session license before it will work. Until you've registered it, you can't connect to the desktop using a PCoIP client.

You receive a registration code when you purchase a pool of licenses from Teradici. Each registration code can be used multiple times; each use consumes one license in its pool.

> **Note: Registration code format**
>
> Registration codes look like this: `ABCDEFGH12@AB12-C345-D67E-89FG`

PCoIP agent license registrations are managed automatically by Teradici's Cloud Licensing service. If necessary, you can manage them yourself, using your own locally-installed PCoIP license server instead.

If you need to purchase licenses, contact Teradici.

Troubleshooting Licensing Issues

If you're encountering problems with Teradici licensing, refer to Troubleshooting License Issues.

Using Teradici Cloud Licensing

To use Cloud Licensing, all you need to do is provide a registration code for each PCoIP agent in your deployment (the same registration code can be used multiple times).

To provide the registration code:

SSH into the agent machine, and invoke `pcoip-register-host` with the license registration code and proxy settings if required:

```
pcoip-register-host --registration-code=<registration-code> [--proxy-
server=<proxy-server-address>] [--proxy-port=<proxy-port-number>]
```

Important: Allowlist network blocks for Teradici Cloud Licensing

If you are using Teradici Cloud Licensing, you will need to add the following to your allowlist:

- `teradici.flexnetoperations.com`
- `teradici.compliance.flexnetoperations.com`

If you use an IP-based allowlist, we recommend your IT team add the following network blocks to your allowlist:

- `IPv4: 185.146.155.64/27`
- `IPv6: 2620:122:f005::/56`

Important: Migrating from the previous specification

Previously, our allowlist specification looked like this:

- **Production**: `64.14.29.0/24`
- **Disaster Recovery**: `64.27.162.0/24`

If you have an existing implementation using an IP-based allowlist like this, we recommend you leave it in place until the new allowlist is active and tested.

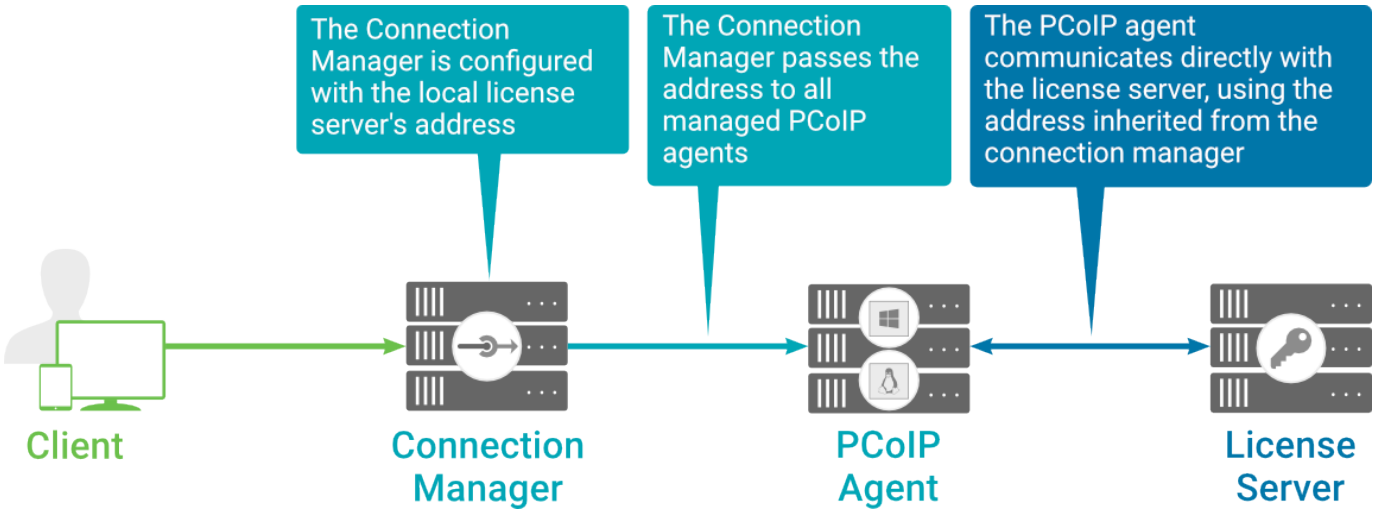
Licensing PCoIP Agents With a Local License Server

In deployments where PCoIP agents cannot access the internet, or where cloud-based licensing is not permitted or desired, a local PCoIP License Server can be used instead. The PCoIP License Server manages PCoIP session licenses within your private environment.

Configuring PCoIP agents to use a local license server is done in one of two ways, depending on whether your deployment uses a PCoIP Connection Manager, or whether your PCoIP clients connect directly to PCoIP agents.

Brokered Environment Licensing

In *brokered* deployments, the license server address is configured in the Connection Manager, which passes it through to its managed PCoIP agents.



Local license validation using a Remote Workstation Card Agent for Linux and a brokered connection

When using a Connection Manager, the license server address is only configured once no matter how many PCoIP agents are behind the Connection Manager.

To set the License Server URL in the Connection Manager:

1. On the Connection Manager machine, use a text editor to open /etc/ConnectionManager.conf.
2. Set the `LicenseServerAddress` parameter with the address of your local license server:
   - `http://{license-server-address}:{port}/request`
3. Save and close the configuration file.
4. Restart the Connection Manager.

VERIFYING YOUR BROKERED LICENSING CONFIGURATION

To verify your system's licensing configuration, run `pcoip-validate-license` from the console on the Remote Workstation Card Agent for Linux machine. The command will ping the license server and attempt to retrieve information on an available license:

```
pcoip-validate-license    license-server-url <license-server-address>[--
server <proxy-server-address>] [--                          ]
```

Where `<license-server-address>` is the address of the license server to ping, formatted as `http://{license-server-address}:{port}/request`

If the license server is behind a proxy server, provide the proxy information via the `-` and `-` parameters.

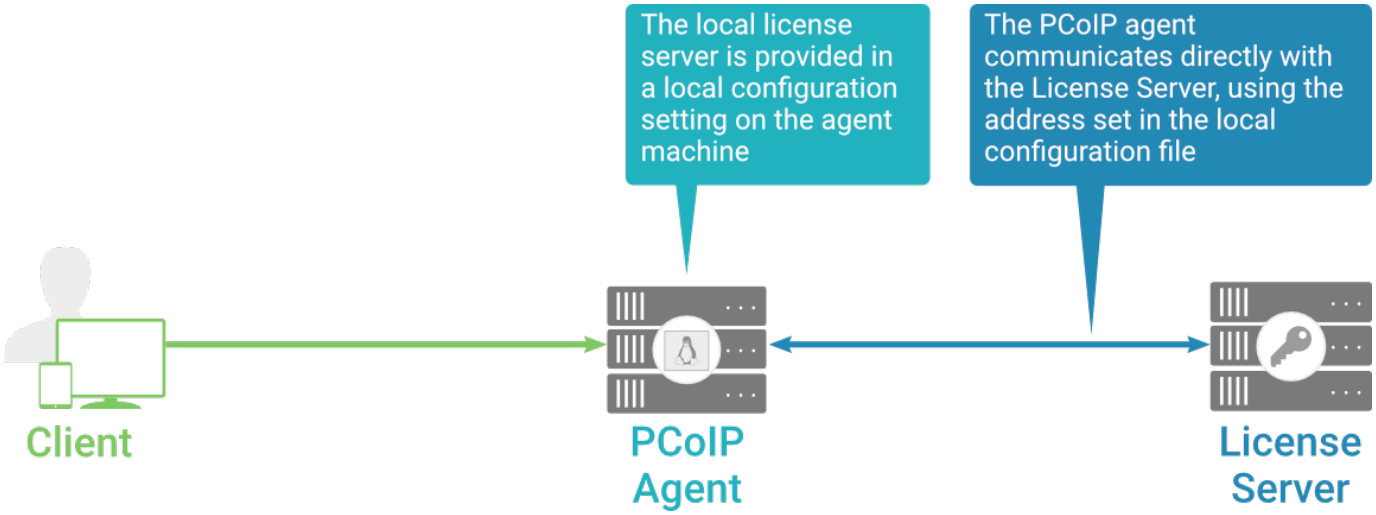
If successful, the response will show that a valid license was found on the license server, and its expiration date.

If the connection is unsuccessful, investigate the following possibilities:

- The license server address is incorrect, or formatted incorrectly.
- The license server is inaccessible.
- There are no available licenses on the license server. `pcoip-validate-license` will only return a positive response if there is at least one available session license.
- If you have only one license on the license server and run `pcoip-validate-license` from a PCoIP session, the command will fail because you are currently using the single license. In this scenario, disconnect your PCoIP session and try again from an SSH session instead.

Unbrokered Environment Licensing

In direct, or unbrokered, deployments, each PCoIP agent is configured with the license server address via a local agent setting. When a client initiates a new PCoIP session, the PCoIP agent uses its local configuration to communicate with the license server.



Local license validation using a Remote Workstation Card Agent for Linux and a direct (unbrokered) connection

Each PCoIP agent in your environment must be individually configured with the license server's URL.

To configure the License Server URL on the Remote Workstation Card Agent for Linux machine:

1. Using a text editor, open `/etc/pcoip-agent/pcoip-agent.conf`.

2. Add or modify the `pcoip.license_server_path` directive:

```
pcoip.license_server_path = <license-server-address>
```

   Where `<license-server-address>` is the address of the license server, formatted as `http://{license-server-address}:{port}/request`.

3. If the license server is behind a proxy server, provide the proxy information using the `pcoip.license_proxy_server` and `pcoip.license_proxy_port` directives.
4. Save and close `pcoip-agent.conf`.

The changes will take effect on the next PCoIP session.

VERIFYING YOUR UNBROKERED LICENSING CONFIGURATION

To verify your system's licensing configuration, run `pcoip-validate-license` from the console on the Remote Workstation Card Agent for Linux machine. The command will ping the license server and attempt to retrieve information on an available license:

```
pcoip-validate-license    license-server-url <license-server-address>[--
server <proxy-server-address>] [--                              ]
```

Where `<license-server-address>` is the address of the license server to ping, formatted as `http://{license-server-address}:{port}/request`

If the license server is behind a proxy server, provide the proxy information via the `-` and `-` parameters.

If successful, the response will show that a valid license was found on the license server, and its expiration date.

If the connection is unsuccessful, investigate the following possibilities:

- The license server address is incorrect, or formatted incorrectly.
- The license server is inaccessible.
- There are no available licenses on the license server. `pcoip-validate-license` will only return a positive response if there is at least one available session license.
- If you have only one license on the license server and run `pcoip-validate-license` from a PCoIP session, the command will fail because you are currently using the single license. In this scenario, disconnect your PCoIP session and try again from an SSH session instead.

Updating the Remote Workstation Card Agent for Linux on RHEL or CentOS

Updates to the Remote Workstation Card Agent for Linux will be published on a regular basis. New stable builds will be produced approximately every three months.

To upgrade to the latest version, use the following three commands:

```
sudo yum makecache
sudo yum update pcoip-agent-standard
sudo reboot
```

Uninstalling the Remote Workstation Card Agent for Linux

You can remove the Remote Workstation Card Agent for Linux from your system, or you can remove the repo config entirely.

Remove the Remote Workstation Card Agent for Linux package

To remove the package, open a console window and run the following command:

```
sudo yum remove pcoip-agent-*
```

Remove the repo configuration

If you want to remove the repo configuration completely, you can do that as well. You'll need to do this if you are switching from one channel to another (for example, from beta to stable), before reconfiguring with the new repo:

```
rm /etc/yum.repos.d/pcoip-agent.repo
rm /etc/yum.repos.d/pcoip-agent-source.repo
```

Configuration Guide

Configuring the PCoIP Agent

You can configure the PCoIP agent, and optimize PCoIP protocol behavior for local network conditions, by adjusting configuration directives found in `/etc/pcoip-agent/pcoip-agent.conf`.

You can find detailed information and descriptions about each setting in the next section. You can also consult the man pages for `pcoip-agent.conf`:

```
man pcoip-agent.conf
```

Only the settings documented here apply to the Remote Workstation Card Agent for Linux

The Remote Workstation Card Agent for Linux man pages document additional configuration settings, beyond those described here. These additional settings apply to virtual machine instances and have no effect on Remote Workstation Card systems. Only the settings described here apply to the Remote Workstation Card.

Applying Configuration Changes

To set or change a configuration value, add or modify directives in `pcoip-agent.conf`. Place one directive on each line, in this format:

```
directive.name = <value>
```

A complete list of configurable values is shown next in Configurable Settings.

Configurable Settings

The following settings can be configured on the Remote Workstation Card Agent for Linux. Refer to Configuring the PCoIP agent to understand how to modify these settings.

License server URL

Directive	Options	Default
`pcoip.license_server_path`	string *(up to **511* characters)*	—

This setting takes effect when you start the next session. This policy sets the license server path. Enter the license server path in 'https://address:port/request' or 'http://address:port/request' format.

PCoIP Security Certificate Settings

Directive	Options	Default
`pcoip.ssl_cert_type`	**1**—From certificate storage **2**—Generate a unique self-signed certificate **0**—From certificate storage if possible, otherwise generate	—
`pcoip.ssl_cert_min_key_length`	**1024**—1024 bits **2048**—2048 bits **3072**—3072 bits **4096**—4096 bits	—

This setting takes effect when you start the next session. A certificate is used to secure PCoIP related communications. The way PCoIP components choose a certificate is based on the certificate type and the key length. Without a certificate being generated or selected, a PCoIP Session cannot be established.

Depending on the value chosen for the option, 'How the PCoIP agent chooses the certificate...' and the availability of appropriate certificates, PCoIP components may acquire a CA signed certificate from certificate storage or generate an in-memory self-signed certificate.

In order for a CA signed certificate to be loadable by PCoIP components, it must be stored at */etc/pcoip-agent/ssl-certs* in three .pem files, owned by the pcoip user, only readable by the owning user.

- pcoip-key.pem must contain an unlocked RSA key
- pcoip-cert.pem must contain a certificate that signs the key in pcoip.pem
- pcoip-cacert.pem must contain a CA certificate chain that validates the certificate in pcoip-cert.pem.

Note: Self-signed certificates are 3072 bits long.

Select a minimum key length (in bits) for a CA signed certificate. Longer length certificates will require more computing resources and may reduce performance, but will increase security. Shorter length certificates will provide better performance at the cost of lower security.

Note: Please refer to Teradici documentation for instructions on creating and deploying certificates.

PCoIP Security Settings

Directive	Options	Default
`pcoip.tls_security_mode`	**0**—Maximum Compatibility	—
`pcoip.tls_cipher_blacklist`	string *(up to **1023* characters)*	—

This setting takes effect when you start the next session. Controls the cryptographic cipher suites and encryption ciphers used by PCoIP endpoints.

The endpoints negotiate the actual cryptographic cipher suites and encryption ciphers based on the settings configured here. Newer versions of TLS and stronger cipher suites will be preferred during negotiation between endpoints.

If this setting is not configured or disabled, the TLS Security Mode will be set to Maximum Compatibility.

TLS Security Mode

Maximum Compatibility offers TLS 1.1, 1.2 and a range of cipher suites including those that support Perfect Forward Security (PFS) and SHA-1. Supported cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_AES_256_GCM_SHA384

Blacklisted Cipher Suites

Provides the ability to block specific cipher suites from being offered during negotiation. Must be entered as a semi-colon separated list of cipher suites.

PCoIP event log verbosity

Directive	Range	Increment	Default
`pcoip.event_filter_mode`	0 – 3	1	2

This setting takes effect immediately. Configures the PCoIP event log verbosity ranging from 0 (least verbose) to 3 (most verbose).

Proxy Access to a remote License Server

Directive	Options	Range	Increment	Default
`pcoip.license_proxy_server`	string *(up to **511* characters)*			—
`pcoip.license_proxy_port`		0 – 65535	1	—

This setting takes effect when you start the next session. If a proxy is required to access a local License Server or the Cloud License Server, enter those parameters here. These parameters are loaded only during agent startup.

X server remote access

Directive	Options	Default
`pcoip.allow_x_remoting`	**0** (off), **1** (on)	—

This setting takes effect when you restart the agent. Configuring this allows you to enable or disable remote access to the X server run by the PCoIP Agent. When not configured, remote access is disabled by default.

Security Guide

PCoIP requires a certificate to establish a session. By default, PCoIP agents generate a self-signed certificate that secures the PCoIP session. Each component in the PCoIP system can generate these self-signed certificates, which will automatically work together without requiring any configuration.

You can, if needed, create and deploy your own custom certificates instead of relying on Teradici's self-signed certificates. This section explains how to create and implement custom certificates.

Using Custom Security Certificates

You can use OpenSSL, Microsoft Certification Authority, or a public certificate authority (CA) of your choice to create your certificates. If you are not using OpenSSL, consult your certificate authority's documentation for instructions on creating certificates in a Windows Certificate Store-compatible format.

The procedures is this section use OpenSSL to generate certificates that will satisfy most security scanner tools when the root signing certificate is known to them.

> ⚠ **Caution: Certificates are stored in the Windows Certificate Store**
>
> Certificates are stored in the Windows certificate store. If you have old certificates that are stored on the host, they should be deleted to avoid conflicts or confusion.

Custom Certificate Guidelines

If you choose to use your own certificates, follow these general guidelines:

- Save your root CA signing certificate in a safe place for deployment to clients.
- Back up private and public keys to secure locations.
- Never store files created when generating keys or certificates on network drives without password protection.

- Once certificates have been deployed to the Windows certificate store, the files they came from are no longer needed and can be deleted.
- Standard automatic tools, such as Automatic Certificate Enrollment and Group Policy, can be used for deploying automatically generated certificates. Both Automatic Certificate Enrollment and Group Policies are implemented through Active Directory. See MSDN Active Directory documentation for more information.

Pre-session Encryption Algorithms

Connections are negotiated using the following supported RSA cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_AES_256_GCM_SHA384

> **Note: Minimum SSL version**
>
> These Max Compatibility security level cipher suites have a minimum required SSL version of TLS 1.2.

Creating And Installing Custom Certificates

This section describes how to replace Teradici's default certificates with your own custom certificates.

> **Note: These procedures use OpenSSL**
>
> The procedures in this section use OpenSSL to create private keys, certificate signing requests, and certificates. To use OpenSSL, install Visual C++ 2008 Restributables and Win32 OpenSSL Light v1.0.2g+.
>
> For detailed information about OpenSSL, refer to OpenSSL documentation.

To replace Teradici's default certificates with custom certificates:

1. Install required OpenSSL components on your system.
2. Create the internal root CA certificate.
3. Create a private key and certificate pair for the PCoIP Agent.
4. Install the agent's private key and certificate in the Windows Certificate Store for each desktop.
5. Configure the certificate mode for each desktop.
6. Install the internal root CA in your PCoIP clients.

Installing OpenSSL Requirements

Install the following components on your Windows machine:

- Visual C++ 2008 Redistributables
- Win32 OpenSSL v1.0.2g Light (or later).

  When prompted during OpenSSL installation, copy the OpenSSL DLLs to the OpenSSL binaries directory; for example, C:\OpenSSL-Win32\bin.

> **Note: Examples use the default installation directory**
>
> The following examples assume the default OpenSSL installation directory: `C:\OpenSSL-Win32`.

Creating the Internal Root CA Certificate

This section shows how to create a root CA private key, how to use this key to self-sign and generate an internal root CA certificate, and how to add X.509 v3 extensions to a certificate that restrict how the certificate can be used.

Creating a Root CA Private Key

To create a root CA private key in RSA format:

1. Open a command prompt (cmd) and navigate to the OpenSSL binaries directory (`c:\OpenSSL-Win32\bin`).
2. Type `openssl` and press `Enter` to launch OpenSSL.

> **Note: OpenSSL may need help finding the .cfg file**
>
> If you see this error:
>
> ```
> WARNING: can't open config file: /usr/local/ssl/openssl.cnf
> ```
>
> you will need to set the OPENSSL_CONF variable before proceeding.

1 To create 3072-bit root RSA key named rootCA.key, use one of the following commands:

```
* For an unsecured key, type:

```bash
genrsa -out rootCA.key 3072
```

* For a password-protected key, add the `-aes128` or `-aes256` argument:

```bash
genrsa -out rootCA.key 3072 -aes256
```

Password-protected keys require the password to be entered each time they are used.
```

> **Caution: Store your private root key in a safe location**
>
> Anyone with access to your private root key can use it to generate certificates that your PCoIP clients will accept.

Setting the OPENSSL_CONF variable

If OpenSSL is unable to find its configuration file, you may need to set the OPENSSL_CONF variable.

To set the OPENSSL_CONF variable:

1. Exit OpenSSL.
2. Type the following command:

```
set OPENSSL_CONF=C:\OpenSSL-Win32\bin\openssl.cfg
```

3. Type `ssl` and press `Enter` to continue with the step you were performing when you saw the error.

Self-signing and Creating the Internal Root CA Certificate

Now that we have our private key, we will use it to generate a self-signed X.509 root CA certificate called **rootCA.pem** that is valid for 1095 days (1095 days is three years, ignoring leap days).

To create the root CA certificate:

1. Type the following command. This command creates a certificate that is valid for 3 years (1095 days). Customize the `-days` parameter to customize the certificate lifetime:

```
req -x509 -new -nodes -key rootCA.key -days 1095 -out rootCA.pem
```

   An interactive script will run, which prompts you to enter values for several fields.Follow the prompts to enter field values:Country NameOptional. Use one of the ISO 3166-1 alpha-2 country codes. State or Province NameOptionalLocality nameOptionalOrganization NameOptionalCommon nameRequired. Enter a name for your root CA (for example, certificates.mycompany.com)Email addressOptional. Enter an administrative alias email if you use this field. Note: Field values can be templatizedIf you will be creating a lot of certificates, consider using a configuration file that contains global field values. See http://www.openssl.org/docs for more information.

Troubleshooting and Support

Support

Contacting Support

If you encounter any problems installing, configuring, or running the Remote Workstation Card Agent for Linux, you can create a support ticket with Teradici.

Before creating a ticket, be prepared with the following:

- A detailed description of the problem
- Your agent version number (how do I find my version number?)
- A prepared support file

The Teradici Community Forum

The PCoIP Community Forum enables users to have conversations with other IT professionals to learn how they resolved issues, find answers to common questions, have peer group discussions on various topics, and access the PCoIP Technical Support Service team. Teradici staff are heavily involved in the forums.

To visit the Teradici community, go to https://communities.teradici.com.

Finding the Agent Version Number

To find the agent's version number in Ubuntu:

```
dpkg -l "pcoip*"
```

To find the agent's version number in RHEL or CentOS:

```
rpm -qai "pcoip*"
```

The console will display a table of all registered PCoIP components and their version number, if they have one.

Creating a Technical Support File

Teradici may request a support file from your system in order to troubleshoot and diagnose PCoIP issues. The support file is an archive containing PCoIP Remote Workstation Card Agent for Linux logs and other diagnostic data that can help support diagnose your problem.

To create a support file, type the following command as a super user:

```
sudo pcoip-support-bundler
```

The support file will be created and placed in your `/tmp` directory. A message will display containing the full system path to the generated file.

Troubleshooting

Performing Diagnostics

Each PCoIP component creates and updates a log file which records its activity as the system is used. Most troubleshooting within a PCoIP system begins by examining these log files and looking for error conditions or other indications that may explain why the system is not operating as expected.

Log files for the Remote Workstation Card Agent for Linux and other PCoIP components are saved to specific directories.

> **Note: Bundling log files for support**
>
> When investigating issues with Teradici support, you may need to provide a support file which includes system log files. Instructions are provided here.

Locating Agent Log Files

Log files for the PCoIP agent are located in the following directories by default. If you changed your agent's location during installation, the log files will be in your custom location instead.

Component	Log file location
Agent	`/var/log/pcoip-agent/agent.log`
Session Launcher	`/var/log/pcoip-agent/session-launcher.log`
Server/User	`/var/log/pcoip-agent/server.<user>.log`

> **Note: Bundling log files for support**
>
> When investigating issues with Teradici support, you may need to provide a support file which includes system log files. Instructions are provided here.

Setting Log Verbosity

Each PCoIP component generates diagnostic log messages. The default log levels are recommended for use in a production deployment. When troubleshooting a particular problem, Teradici Support Services may recommend adjusting the PCoIP event log verbosity level to obtain more information from certain parts of the system.

> **Note: This is a global setting**
>
> The `pcoip.event_filter_mode` directive is a global setting, and affects the output levels of all PCoIP components.

To change the log verbosity level, set the `pcoip.event_filter_mode` directive in the `pcoip-agent.conf` file. See Configuring the PCoIP Agent for instructions.

Log rotation

Log files in Linux agents are managed by `logrotate`. To manage how log files are rotated, edit the following files:

- `/etc/logrotate.d/pcoip-*`
- `/usr/share/pcoip-agent/pcoip-server.logrotate`

Session Log IDs

At the start of each PCoIP session, a unique session ID is generated by the PCoIP Client and passed to all connected PCoIP components (including the Remote Workstation Card Agent for Linux). Log messages generated by the agent are prefixed with this session ID, making it easy to identify. All log messages generated during a single session, by any PCoIP component, will be prefixed with the same session log ID in RFC-4122 format:

```
yyyy-mm-ddThh:mm:ss.ffffffZ xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx > …
```

For example:

```
2015-11-06T08:01:18.688879Z 4208fb66-e22a-11d1-a7d7-00a0c982c00d > …
```

Log messages that do not pertain to a specific session will show a string of zeroes in place of the session log ID number.

If a PCoIP component does not receive a session log ID from the PCoIP client, or receives an invalid value, it will generate a new session log ID and distribute it to the other components in the system.

Troubleshooting License Issues

Teradici includes a license validation utility that scans your local system and any connected physical or cloud-based license servers for active licenses, and informs you of when your license subscription expires. For more information, see FAQ - Licensing HP Anyware in our Knowledge Base.

To run the license validation tool, type:

```
pcoip-validate-license
```

For more detailed information on `pcoip-validate-license`, type:

```
man pcoip-validate-license
```

To list your licenses and their expiration status, type:

```
pcoip-list-licenses
```

For more detailed instructions on `pcoip-list-licenses`, type:

```
man pcoip-list-licenses
```

Tracking Usage Over Time

Teradici Local License Server users can use our open-source script, which displays the maximum HP Anyware license concurrent usage for a license server over time. For more information, refer to our Github page.

Teradici Cloud Licensing users can write a short script that runs `pcoip-list-licenses` periodically (for example, every 60 minutes) on any PCoIP agent machine to track license usage.

Frequently Asked Questions

Can I use a screensaver?

Yes. However, a blank, static screensaver will provide the most efficient CPU and network bandwidth usage.

How quickly does a PCoIP agent complete a connection?

PCoIP agents can usually achieve a connection in 15 to 30 seconds. We use the statistical value Top Percentile (TP) to measure the time to establish a session:

- TP99: Ninety-nine percent of connections complete in under 30 seconds.
- TP50: Fifty percent of connections complete in under 15 seconds.

Why is my application not sending audio?

The PCoIP agent delivers audio over PCoIP connections by reassigning the system's default audio device. Only applications that use the system default audio device will send or receive audio over PCoIP; applications that are configured to use non-default devices will not work. If you don't hear audio from your application, make sure it is configured to use the system default audio device.

I'm using Teradici Cloud Licensing. What network blocks should I leave open?

If you are using Teradici Cloud Licensing, you will need to add the following to your allowlist:

- `teradici.flexnetoperations.com`
- `teradici.compliance.flexnetoperations.com`

If you use an IP-based allowlist, we recommend your IT team add the following network blocks to your allowlist:

- `IPv4: 185.146.155.64/27`
- `IPv6: 2620:122:f005::/56`

Important: Migrating from the previous specification

Previously, our allowlist specification looked like this:

- **Production**: `64.14.29.0/24`
- **Disaster Recovery**: `64.27.162.0/24`

If you have an existing implementation using an IP-based allowlist like this, we recommend you leave it in place until the new allowlist is active and tested.